Exercise 1: Configuring a Basic Spring Application

# Scenario:

Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations.

# 1. Set Up a Spring Project

a. Create a Maven project named LibraryManagement.

b. Add Spring Core dependencies in the pom.xml file:

<project xmlns="http://maven.apache.org/POM/4.0.0"   
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0   
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.library</groupId>  
 <artifactId>LibraryManagement</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <dependencies>  
 <!-- Spring Core Dependency -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.31</version>  
 </dependency>  
 </dependencies>  
</project>

# 2. Configure the Application Context

a. Create applicationContext.xml in src/main/resources:

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="  
 http://www.springframework.org/schema/beans   
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <!-- Define BookRepository bean -->  
 <bean id="bookRepository" class="com.library.repository.BookRepository" />  
  
 <!-- Define BookService bean and inject BookRepository -->  
 <bean id="bookService" class="com.library.service.BookService">  
 <property name="bookRepository" ref="bookRepository" />  
 </bean>  
</beans>

# 3. Define Service and Repository Classes

a. Create BookRepository.java in com.library.repository:

package com.library.repository;  
  
public class BookRepository {  
 public void saveBook(String bookName) {  
 System.out.println("Book saved: " + bookName);  
 }  
}

b. Create BookService.java in com.library.service:

package com.library.service;  
  
import com.library.repository.BookRepository;  
  
public class BookService {  
 private BookRepository bookRepository;  
  
 // Setter injection  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void addBook(String bookName) {  
 System.out.println("Adding book: " + bookName);  
 bookRepository.saveBook(bookName);  
 }  
}

# 4. Run the Application

a. Create MainApp.java in com.library:

package com.library;  
  
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class MainApp {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 BookService bookService = (BookService) context.getBean("bookService");  
 bookService.addBook("The Great Gatsby");  
 }  
}

# Expected Output:

Adding book: The Great Gatsby  
Book saved: The Great Gatsby